

LLL		NNN	NNN	KKK	KKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	KKK	KKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	KKK	KKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	KKK	KKK	EEE	RRR
LLL		NNNNNN	NNN	KKK	KKK	EEE	RRR
LLL		NNNNNN	NNN	KKK	KKK	EEE	RRR
LLL		NNNNNN	NNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	NNN	KKKKKKKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	NNN	KKKKKKKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	NNN	KKKKKKKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNNNNN	KKK	KKK	EEE	RRR
LLL		NNN	NNNNNN	KKK	KKK	EEE	RRR
LLL		NNN	NNNNNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	KKK	KKK	EEE	RRR
LLL		NNN	NNN	KKK	KKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	KKK	KKK	EEEEEEEEE	RRRRRRRRRR
LLL		NNN	NNN	KKK	KKK	EEEEEEEEE	RRRRRRRRRR

\*\*FILE\*\*ID\*\*LNKPROSHR

LL	NN	NN	KK	KK	PPPPPPPP	RRRRRRRR	000000	SSSSSSSS	HH	HH	RRRRRRRR	
LL	NN	NN	KK	KK	PPPPPPPP	RRRRRRRR	000000	SSSSSSSS	HH	HH	RRRRRRRR	
LL	NN	NN	KK	KK	PP	PP	RR	RR	00	00	RR	
LL	NN	NN	KK	KK	PP	PP	RR	RR	00	00	RR	
LL	NNNN	NN	KK	KK	PP	PP	RR	RR	00	00	RR	
LL	NNNN	NN	KK	KK	PP	PP	RR	RR	00	00	RR	
LL	NN	NN	NN	KKKKKK	PPPPPPPP	RRRRRRRR	00	00	SSSSSS	HHHHHHHHHH	RRRRRRRR	
LL	NN	NN	NN	KKKKKK	PPPPPPPP	RRRRRRRR	00	00	SSSSSS	HHHHHHHHHH	RRRRRRRR	
LL	NN	NNNN	KK	KK	PP	RR	RR	00	00	SS	HH	
LL	NN	NNNN	KK	KK	PP	RR	RR	00	00	SS	HH	
LL	NN	NN	KK	KK	PP	PP	RR	RR	00	00	RR	
LL	NN	NN	KK	KK	PP	PP	RR	RR	00	00	RR	
LLLLLLLLLL	NN	NN	KK	KK	PP	RR	RR	000000	SSSSSSSS	HH	HH	RR
LLLLLLLLLL	NN	NN	KK	KK	PP	RR	RR	000000	SSSSSSSS	HH	HH	RR

```
1 0001 0 module lnk_procshrim ! PROCESS SHAREABLE IMAGES ON PASS 1
2 0002 0 (ident = 'V04-000'
3 0003 0 ,addressing mode
4 0004 0 (external = general
5 0005 0 ;nonexternal = long_relative
6 0006 0
7 0007 0 )
8 0008 1 begin
9 0009 1 ****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 * ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *
31 0031 1 ****
32 0032 1 *
33 0033 1 *
34 0034 1 ++
35 0035 1 FACILITY: LINKER
36 0036 1 ABSTRACT: THIS MODULE CONTAINS THE ROUTINES TO READ SHAREABLE
37 0037 1 IMAGES ON PASS 1.
38 0038 1 *
39 0039 1 *
40 0040 1 ENVIRONMENT: VMS NATIVE MODE
41 0041 1 *
42 0042 1 AUTHOR: T.J. PORTER, CREATION DATE: 15-DEC-77
43 0043 1 *
44 0044 1 MODIFIED BY:
45 0045 1 *
46 0046 1 V03-008 ADE0003 Alan D. Eldridge 7-Aug-1984
47 0047 1 Propagate the DZRO bit from the shareable image ISD to the
48 0048 1 corresponding ISD in the new image. It was being cleared
49 0049 1 unconditionally.
50 0050 1 *
51 0051 1 V03-007 JWT0192 Jim Teague 2-Aug-1984
52 0052 1 Unfortunately, push came to shove, and the global
53 0053 1 isd search optimization has been tossed.
54 0054 1 *
55 0055 1 V03-006 JWT0189 Jim Teague 25-Jul-1984
56 0056 1 Make some changes to accomodate demand-zero image
57 0057 1 *
```

```

58      0058 1 | sections in shareable images. The minimum image
59      0059 1 | file VBN for the symbol table is now 2, instead of 3.
60      0060 1 | Why? Consider the case of a shareable image with
61      0061 1 | nothing BUT a demand zero image section...
62      0062 1 |
63      0063 1 | V03-005 ADE0002 Alan D. Eldridge 10-Jul-1984
64      0064 1 | Fix null arguments passed in SIGNAL due to editting errors.
65      0065 1 |
66      0066 1 | V03-004 ADE0001 Alan D. Eldridge 26-Feb-1984
67      0067 1 | Don't erase CLUSL_GSMATCH. It is no longer re-used as
68      0068 1 | something else in pass 2, and it is referenced to correctly
69      0069 1 | perform the GSMATCH checking in pass one.
70      0070 1 |
71      0071 1 | V03-003 JWT0152 Jim Teague 8-Feb-1984
72      0072 1 | Long names for global image section descriptors.
73      0073 1 |
74      0074 1 | V03-002 JWT0111 Jim Teague 14-Apr-1983
75      0075 1 | Don't pull in shareable images that are in global
76      0076 1 | isds unless they're based.
77      0077 1 |
78      0078 1 | V03-001 JWT0044 Jim Teague 30-Jul-1982
79      0079 1 | Open file performance boost.
80      0080 1 |
81      0081 1 | --
82      0082 1 |
83      0083 1 |
84      0084 1 | TABLE OF CONTENTS:
85      0085 1 |
86      0086 1 | forward routine
87      0087 1 |   readnextblock; ! READS THE NEXT HEADER BLOCK OF IMAGE
88      0088 1 |
89      0089 1 |
90      0090 1 | INCLUDE FILES:
91      0091 1 |
92      0092 1 | library 'LIBL32';
93      0093 1 | require 'PREFIX';
94      0208 1 | library 'DATBAS';
95      0209 1 | require 'ISGENC';
96      0593 1 |
97      0594 1 |
98      0595 1 | MACROS:
99      0596 1 |
100     0597 1 |   NONE
101     0598 1 |
102     0599 1 | EQUATED SYMBOLS:
103     0600 1 |
104     0601 1 |   NONE
105     0602 1 |
106     0603 1 | EXTERNAL REFERENCES:
107     0604 1 |
108     0605 1 |
109     0606 1 | external routine
110     0607 1 |   lnk$alloblk : novalue, ! DYNAMIC MEMORY ALLOCATOR
111     0608 1 |   lnk$alloccluster, : novalue, ! ALLOCATE CLUSTER DESCRIPTOR
112     0609 1 |   lnk$closefile : novalue, ! CLOSE CURRENT INPUT FILE
113     0610 1 |   lnk$pointobj : novalue, ! POINTS TO NEW PLACE IN FILE
114     0611 1 |   lnk$addimage, ! PROCESS SHAREABLE IMAGE

```

```

: 115      0612 1   Lnk$procsobj;
: 116      0613 1
: 117      0614 1   external literal
: 118      0615 1     lins$badimghdr,
: 119      0616 1     lins$basshrbel,
: 120      0617 1     lins$confbasadr,
: 121      0618 1     lins$idmismch,
: 122      0619 1     lins$imgbased,
: 123      0620 1     lins_noimgfil,
: 124      0621 1     lins_nonpicimg,
: 125      0622 1     lins_relink,
: 126      0623 1     lins_readerr;
: 127      0624 1
: 128      0625 1   external
: 129      0626 1     lnk$gl_shrclstrs,
: 130      0627 1     lnk$al_rab : block [, byte],
: 131      0628 1     lnk$gw_nisects : word,
: 132      0629 1     lnk$gw_shriscts : word,
: 133      0630 1     lnk$gb_pass : byte,
: 134      0631 1     lnk$gl_curfil : ref block [, byte],
: 135      0632 1     lnk$gl_curclu : ref block [, byte],
: 136      0633 1     lnk$gl_defclu : block [, byte],
: 137      0634 1     lnk$gl_lastclu : ref block [, byte],
: 138      0635 1     lnk$gl_ctlmsk : block [, byte],
: 139      0636 1     lnk$gl_objrecs : block [, byte];
: 140      0637 1
: 141      0638 1   ! MODULE OWN STORAGE:
: 142      0639 1
: 143      0640 1
: 144      0641 1   global
: 145      0642 1     lnk$gl_gsbufdsc : vector [3],           ! BUFFER DESCRIPTOR FOR COPIED GLOBAL SECTIONS
: 146      0643 1     lnk$gl_imgrecs ;                      ! COUNT OF IMAGE RECORDS
: 147      0644 1
: 148      0645 1   own
: 149      0646 1     curisdseq : vector [4, byte]           ! SUFFIX TO SCLUSTER NAME GIVING GBL ISD NAME
: 150      0647 1     initial (%ascii '_000'),          ! NUMBER REMAINING HEADER BLOCKS
: 151      0648 1     hdrblkcnt,                         ! CURRENT HEADER VBN
: 152      0649 1     headerblock ;
: 153      0650 1
: 154      0651 1

```

! PROCESSES OBJECT MODULES (I.E. THE GST)

BAD IMAGE HEADER ERROR MESSAGE  
BASED SHAREABLE IMAGE BELOW BASE=  
CONFLICTING BASE ADDRESSES FOR SHR IMGS  
GSMATCH MISMATCH WITH SHR IMG STB LIBRARY  
ATTEMPT TO RE-BASE A NON-PIC IMAGE  
NO IMAGE FILE CREATED  
SHAREABLE IMAGE IS NON-PIC  
RELINK DUE TO COPYALWAYS SECTION  
READ ERROR

COUNT OF NUMBER OF SHAREABLE IMAGE CLUSTERS (PIC AND NON-P  
RAB FOR OPEN IMAGE FILE  
IMAGE SECTION ACCUMULATOR  
NUMBER OF SHAREABLE IMAGE ISECTS  
CURRENT PASS  
CURRENT FILE POINTER  
CURRENT CLUSTER DESCRIPTOR  
DEFAULT CLUSTER DESCRIPTOR  
POINTER TO LAST CLUSTER DESCRIPTOR  
CONTROL FLAGS  
COUNT OF OBJECT RECORDS

! BUFFER DESCRIPTOR FOR COPIED GLOBAL SECTIONS  
! COUNT OF IMAGE RECORDS

! SUFFIX TO SCLUSTER NAME GIVING GBL ISD NAME  
! NUMBER REMAINING HEADER BLOCKS  
! CURRENT HEADER VBN

```

: 156      0652 1 global routine Lnk$procshrim (modrfa) =      !
: 157      0653 1 ++
: 158      0654 1 FUNCTIONAL DESCRIPTION:
: 159      0655 1      THIS ROUTINE IS CALLED TO PROCESS SHAREABLE IMAGES ON PASS
: 160      0656 1      ONE OF THE LINK. IT READS AND VALIDATES EVERY BLOCK OF THE IMAGE HEADER
: 161      0657 1      BUILDING THE LIST OF IMAGE SECTION DESCRIPTORS. AFTER COMPLETING THE HEADER
: 162      0658 1      IT SETS THE RAB POINTING TO THE SYMBOL TABLE PART OF THE IMAGE FILE AND
: 163      0659 1      CALLS LNK$PROCSOBJ TO DO THE PASS ONE OBJECT MODULE PROCESSING
: 164      0660 1      OF THE SYMBOL TABLE.
: 165      0661 1      THE SYMBOL TABLE OF A SHAREABLE IMAGE CONTAINS ALL THE UNIVERSAL
: 166      0662 1      SYMBOLS DEFINED FOR THE IMAGE AND IS IN THE OBJECT MODULE FORMAT.
: 167      0663 1
: 168      0664 1 FORMAL PARAMETERS:
: 169      0665 1      MODRFA IS A POINTER TO THE 6 BYTE RFA OF THE SHAREABLE IMAGE IF
: 170      0666 1      IT IS IN A LIBRARY. IF NOT A LIBRARY SHAREABLE IMAGE THE ARGUMENT
: 171      0667 1      DOES NOT EXIST.
: 172      0668 1
: 173      0669 1 IMPLICIT INPUTS:
: 174      0670 1      SEE ABOVE EXTERNAL DECLARATIONS.
: 175      0671 1      IN ADDITION THE FILE CONTAINING THE IMAGE IS ALREADY OPEN
: 176      0672 1      FOR BLOCK READ OPERATIONS.
: 177      0673 1
: 178      0674 1 IMPLICIT OUTPUTS:
: 179      0675 1      SEE ABOVE GLOBAL DECLARATIONS.
: 180      0676 1      IN ADDITION ALL IMAGE SECTIONS FOUND IN THIS SHAREABLE IMAGE
: 181      0677 1      ARE APPENDED TO THE LIST AND THE GST HAS BEEN PROCESSED AS AN OBJECT
: 182      0678 1      MODULE. THAT IS ALL UNIVERSAL SYMBOLS ARE IN THE LINKER SYMBOL TABLE.
: 183      0679 1
: 184      0680 1 ROUTINE VALUE:
: 185      0681 1      RETURNS VALUE TRUE IF SUCCESSFULLY PROCESSED, ELSE FALSE
: 186      0682 1
: 187      0683 1 SIDE EFFECTS:
: 188      0684 1      THE ROUTINE DOES NOT RETURN IF A FATAL ERROR IS DETECTED.
: 189      0685 1
: 190      0686 1 --
: 191      0687 2 begin
: 192      0688 2 builtin
: 193      0689 2 actualcount :          ! GETS COUNT OF ARGUMENTS
: 194      0690 2 local
: 195      0691 2 nxtisoff,          ! OFFSET TO NEXT ISD
: 196      0692 2 blockoffset,       ! OFFSET IN FILE TO FIRST HEADER BLOCK
: 197      0693 2 isectident,        ! MAJOR AND MINOR ID FROM HEADER
: 198      0694 2 firstisdvpg,       ! VPG OF FIRST ISECT
: 199      0695 2 symdbgdatdsc : ref block [. byte], ! POINTER TO SYMBOL TABLE DESCRIPTOR
: 200      0696 2 gstrecs,          ! NUMBER OF RECORDS IN SYMBOL TABLE
: 201      0697 2 ownclu : ref block [. byte], ! POINTER TO OWNING CLUSTER DESCRIPTOR
: 202      0698 2 ownfdb : ref block [. byte], ! AND IT'S FILE DESCRIPTOR BLOCK
: 203      0699 2 cludesc : ref block [. byte], ! POINTER TO CREATED CLUSTER DESCRIPTOR
: 204      0700 2 curhdrisd : ref block [. byte], ! POINTER TO CURRENT ISD IN HEADER
: 205      0701 2 curisd : ref block [. byte], ! POINTER TO CURRENT ISD BEING BUILT
: 206      0702 2 newhdrisd : ref block [. byte], ! POINTER TO HEADER PART OF CURRENT ISD BEING BUILT
: 207      0703 2 firstisd : ref block [. byte], ! POINTER TO FIRST ISECT IN CLUSTER
: 208      0704 2 gstdvbn : vector [2, {long}], ! RFA OF GST
: 209      0705 2 iafva,           ! RELATIVE VA OF FIXUP SECTION
: 210      0706 2 saverecount,      ! SAVED RECORD COUNT WHILE IN OBJPS1
: 211      0707 2 maxisdvbn ;    ! LAST IMAGE VBN + 1
: 212      0708 2

```

```

: 213      0709 2 map      modrfa : ref block [, byte] ;           ! POINTER TO BLOCK
: 214      0710 2          modrfa : ref block [, byte] ;
: 215      0711 2
: 216      0712 2 bind     header = Lnk$al_rab [rab$l_ubf] : ref block [, byte] ; ! POINTER TO BLOCK BUFFER
: 217      0713 2
: 218      0714 2 if actualcount () eql 0           ! IF CALLED WITH NO ARGUMENTS
: 219      0715 2 then      blockoffset = 0           ! HEADER AT START OF FILE
: 220      0716 2
: 221      0717 2 else      blockoffset = .modrfa [rfa$l_vbn] - 1 ; ! OTHERWISE GET OFFSET.
: 222      0718 2
: 223      0719 2
: 224      0720 2
: 225      0721 2 Lnk$gl_shrclstrs = .Lnk$gl_shrclstrs + 1 ; ! COUNT THIS SHAREABLE IMAGE CLUSTER
: 226      0722 2
: 227      0723 2 cludesc   = 0 ;           ! NO CREATED CLUSTER YET
: 228      0724 2 hdrblkcnt = 1 ;           ! MUST BE AT LEAST ONE BLOCK
: 229      0725 2 nxtisoff  = -1 ;           ! NEXT ISD IS ON NEXT BLOCK
: 230      0726 2 maxisvbn = 0 ;           ! RESET LAST BLOCK OF IMAGE
: 231      0727 2 headerblock = .blockoffset ; ! AND SET FOR FIRST BLOCK READ
: 232      0728 2
: 233      0729 2 ch$fill (%c'0', 3, curisdseq [1]) ; ! INITIALIZE THE ISD NAME SUFFIX
: 234      0730 2
: 235      0731 2 while .nxtisoff eql -1           ! WHILE THERE ARE MORE
: 236      0732 3 do begin
: 237      0733 3     if not readnextblock ()           ! HEADER BLOCKS
: 238      0734 3     then
: 239      0735 3         signal_stop (lin$badimghdr, 1           ! GET THE NEXT ONE
: 240      0736 3             ,lnk$gl_curfil [fdb$g_filename]
: 241      0737 3         );
: 242      0738 4 if .headerblock eql (.blockoffset + 1) ! AND IF UNSUCCESSFUL, FATAL IMAGE
: 243      0739 3 then
: 244      0740 4     begin
: 245      0741 4         ! HAVE SOME VALIDATION TO DO
: 246      0742 4         VALIDATE IMAGE HEADER
: 247      0743 4
: 248      0744 4         if .header [ihd$b_imgtype] neq ihd$k_shr           ! CHECK IT IS A SHAREABLE IMAGE
: 249      0745 4         or .header [ihd$w_majorid] neq ihd$k_majorid       ! MAJOR HEADER ID MUST MATCH
: 250      0746 6         or (.header [ihd$w_minorid]<0, 8, 0> eql (ihd$k_minorid and %x'FF')
: 251      0747 5             and
: 252      0748 5             (.header [ihd$w_minorid]<8, 8, 0> gtru (ihd$k_minorid and %x'FF00')/256
: 253      0749 5         )
: 254      0750 5         or .header [ihd$w_size] gtru maxu ((.header [ihd$w_patchoff] ! THE HEADER FIXED PART
: 255      0751 5             + ihp$k_length) ! MUST BE LESS THAN A BLOCK AND MUST
: 256      0752 4             ,ihd$k_maxlength
: 257      0753 4
: 258      0754 4         or (hdrblkcnt = .header [ihd$b_hdrblkcnt] - 1) lss 0 ! CONTAIN PATCH AREA. 0 TO
: 259      0755 5         or (symdbgdatdsc = .header + .header [ihd$w_symdbgoff]) ! 127 MORE BLOCKS. GET THE
: 260      0756 5             gequ (.header + .header [ihd$w_size]) ! GST DESCRIPTOR WHICH MUST BE CONTAINED
: 261      0757 4         or (gstrecs  = .symdbgdatdsc [ihs$w_gstrecs])-lssu 3 ! IN HEADER. MUST BE AT LEAST 3 RECORDS
: 262      0758 5         or (gstvbn [0]  = .symdbgdatdsc [ihs$1_gstvbn])
: 263      0759 5             lequ (.hdrblkcnt + 1) ! AND MUST BE BEYOND THE HEADER BLOCKS
: 264      0760 4
: 265      0761 4         then
: 266      0762 4             signal_stop (lin$badimghdr, 1           ! ANY ABOVE NOT TRUE, FATAL IMAGE
: 267      0763 4                 ,lnk$gl_curfil [fdb$g_filename] ! HEADER ERROR
: 268      0764 5         if not (lnk$gl_curclu [clu$v_pic] = .header [ihd$v_picimg]) ! EXTRACT THE PIC BIT AND IF NON-PIC
: 269      0765 4         then

```

```

: 270      0766 5      begin
: 271      0767 5      lNK$gl_ctlmsk [lnk$v_picimg] = false ;
: 272      0768 5
: 273      0769 5      if      .lnk$gl_curclu [clu$v_usrbased]
: 274      0770 5      and not .lnk$gl_curclu [clu$v_intclu]
: 275      0771 5      then
: 276      0772 6      begin
: 277      0773 6      signal  (lins_imgbased, 1
: 278      0774 6      ,lnk$gl_curfil [fdb$q_filename]
: 279      0775 6      );
: 280      0776 6      lNK$gl_curclu [clu$v_usrbased] = false ;
: 281      0777 6      lNK$gl_curclu [clu$L_usrbase] = 0 ;
: 282      0778 5      end ;
: 283      0779 5
: 284      0780 4      end
: 285      0781 4      else
: 286      0782 4      if .lnk$gl_curclu [clu$v_usrbased]
: 287      0783 4      then
: 288      0784 4      lNK$gl_ctlmsk [lnk$v_picimg] = false ;
: 289      0785 4      if      .lnk$gl_ctlmsk [lnk$v_shr]
: 290      0786 4      and not .lnk$gl_ctlmsk [lnk$v_ubased]
: 291      0787 4      and not .lnk$gl_curclu [clu$v_pic]
: 292      0788 4      then
: 293      0789 4      signal  (lins_nonpicimg, 1
: 294      0790 4      ,lnk$gl_curfil [fdb$q_filename]
: 295      0791 4      );
: 296      0792 5      if (lnk$gl_curclu [clu$v_matchctl] =
: 297      0793 4      .header [ihd$v_matchctl]) eql isd$k_matnev
: 298      0794 4      then
: 299      0795 4      lNK$gl_curclu [clu$v_copy] = true ;
: 300      0796 4
: 301      0797 4      isectident = .header [ihd$L_ident] ;
: 302      0798 4      iafva = 0 ;
: 303      0799 4
: 304      0800 5      if (.header + .header [ihd$w_activoff])
: 305      0801 4      gtru header [ihd$L_iafva]
: 306      0802 4      then
: 307      0803 5      begin
: 308      0804 5      iafva = .header [ihd$L_iafva] ;
: 309      0805 5      if .iafva eql 0
: 310      0806 5      then
: 311      0807 5      lNK$gl_curclu [clu$v_prefixup] = true ;
: 312      0808 5      end
: 313      0809 4      else
: 314      0810 4      lNK$gl_curclu [clu$v_prefixup] = true ;
: 315      0811 4
: 316      0812 4      lNK$gl_imgrecs = .lnk$gl_imgrecs + .hdrblkcnt + 1 ;
: 317      0813 4      curhdrisd = .header + .header [ihd$w_size] ;
: 318      0814 4
: 319      0815 4
: 320      0816 4      CHECK GSMATCH OF IMAGE AGAINST GSMATCH FOUND IN SHAREABLE IMAGE
: 321      0817 4      LIBRARY IF THIS IMAGE FOUND IN A LIBRARY
: 322      0818 4
: 323      0819 4      if      .lnk$gl_curclu [clu$L_gsmatch] neq 0
: 324      0820 4      and .lnk$gl_curclu [clu$L_gsmatch] neq .isectident
: 325      0821 4      then
: 326      0822 5      begin

```

LN  
VO

```

327      0823 5      bind libident = lnk$gl_curclu [clu$l_gsmatch] : block [, byte],
328      0824 5      imgident = isectident : block [, byte];
329      0825 5
330      0826 5      signal (lin$ idmismch, 6 ! WARN USER ABOUT IDENT PROBLEM
331      0827 5      ,.imgident [gmt$b_majorid],.imgident [gmt$b_minorid]
332      0828 5      ,lnk$gl_curfil [fdb$g_filename], .libident [gmt$b_majorid],.libident [gmt$b_minorid]
333      0829 5      ,lnk$gl_curfil [fdb$g_libnamdsc]
334      0830 5      ;
335      0831 4      end;
336      0832 4      end ! ALL DONE WITH FIRST BLOCK
337      0833 3      else curhdrisd = .header; ! OTHER BLOCKS ARE ALL ISD
338      0834 3
339      0835 3
340      0836 3
341      0837 3      | PROCESS IMAGE SECTION DESCRIPTORS
342      0838 3
343      0839 3      while (nxtisoff = .(curhdrisd [isd$w_size])<0, 16, 1>) gtr 0 ! WHILE MORE ISD'S ON THIS BLOCK
344      0840 4      do begin
345      0841 4          if .curhdrisd [isd$b_type] neq isd$k_usrstack ! BEGIN LOOP THAT PROCESSES EACH
346      0842 4          then
347      0843 5              begin
348      0844 5                  if .curhdrisd [isd$v_copyalway] ! IGNORING STACK SECTION
349      0845 5                  then
350      0846 5                      signal_stop (lin$_relink, 2 ! IF THIS IS COPY ALWAYS SECTION
351      0847 5                      ,lnk$gl_curfil [fdb$g_filename]
352      0848 5                      ,lnk$gl_curfil [fdb$g_filename]
353      0849 5                      );
354      0850 5
355      0851 5          if .curhdrisd [isd$v_gbl] ! IF GLOBAL ISECT
356      0852 5          then
357      0853 6              begin
358      0854 6                  local found, ! THEN CHECK FROM SAME SHAREABLE IMAGE
359      0855 6                  shrdesc : block [dsc$c_s_bln, byte];
360      0856 6                  bind
361      0857 6                      gblnam = curhdrisd [isd$st_gblnam] : vector [, byte];
362      0858 6
363      0859 6                      shrdesc [dsc$a_pointer] = gblnam [1]; ! COMPLETE SECTION NAME DESCRIPTOR
364      0860 6                      shrdesc [dsc$w_length] = .gblnam [0] - suffix_size; ! FIND LENGTH OF GLOBAL SECTION
365      0861 6
366      0862 6          if .cludesc eql 0 ! SECTION NAME IN ISECT DESCRIPTOR
367      0863 6          or not ch$eq (shrdesc [dsc$w_length], gblnam [1]) ! IF NO CREATED CLUSTERS YET
368      0864 6          ,.shrdesc [dsc$w_length], cludesc [clu$st_name] ! OR THIS IS FOR A DIFFERENT CLUSTER
369      0865 6
370      0866 6
371      0867 7      then
372      0868 7          begin
373      0869 7              lnk$addimage (shrdesc, 0, cludesc, found); ! ADD IMAGE TO THE CLUSTER LIST
374      0870 7          if .cludesc neq 0 and not .found
375      0871 8              then
376      0872 8                  begin
377      0873 8                      cludesc [clu$v_prefixup] = .lnk$gl_curclu [clu$v_prefixup]; ! COPY PREFIXUP FLAG FROM PARENT CLUSTE
378      0874 8                      cludesc [clu$l_ownclu] = .lnk$gl_curclu; ! SET OWNING CLUSTER DESCRIPTOR
379      0875 8
380      0876 8          if not .lnk$gl_curclu [clu$v_pic] ! IF THIS IMAGE IS NON-PIC
381      0877 8          then
382      0878 9              begin
383      0879 9                  cludesc [clu$v_based] = true; ! FLAG CLUSTER AS BASED

```

```
384      0880 10
385      0881 10
386      0882 10
387      0883 10
388      0884 9
389      0885 9
390      0886 9
391      0887 8
392      0888 8
393      0889 7
394      0890 8
395      0891 9
396      0892 9
397      0893 9
398      0894 9
399      0895 8
400      0896 8
401      0897 9
402      0898 9
403      0899 9
404      0900 9
405      0901 9
406      0902 9
407      0903 9
408      0904 9
409      0905 9
410      0906 9
411      0907 9
412      0908 9
413      0909 9
414      0910 9
415      0911 9
416      0912 9
417      0913 10
418      0914 10
419      0915 10
420      0916 10
421      0917 10
422      0918 10
423      0919 10
424      0920 10
425      0921 10
426      0922 10
427      0923 9
428      0924 10
429      0925 10
430      0926 10
431      0927 11
432      0928 11
433      0929 11
434      0930 11
435      0931 11
436      0932 11
437      0933 11
438      0934 10
439      0935 9
440      0936 8

        cludesc [clu$l_base] = (if .curhdrisd [isd$v_based]
                                then .curhdrisd [isd$v_vpg]^9
                                else .lnk$gl_curclu [clu$l_base] +
                                     .lnk$gl_curclu [clu$l_cluoff]
                                );
        lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
                                         .curhdrisd [isd$w_pagcnt]*512 ;
      end ;
    else begin
      if (.lnk$gl_curclu [clu$v_based]) ! IF THIS CLUSTER IS BASED
        and not
        .lnk$gl_curclu [clu$v_pic]
      or .lnk$gl_curclu [clu$v_usrbased]
      then begin
        ownclu = .cludesc [clu$l_ownclu] ; ! GET OWNING CLUSTER POINTER
        if .ownclu neq 0
        then
          lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
                                         .curhdrisd [isd$w_pagcnt]*512 ;
        if .ownclu neq 0 ! IF THERE IS AN OWNING CLUSTER
          and
          .ownclu [clu$v_based] ! WHICH IS ALSO BASED
          and
          .cludesc [clu$l_base] neq .curhdrisd [isd$v_vpg]^9
          ! AND WANTS THIS CLUSTER AT A
          ! DIFFERENT PLACE
        then begin
          ownfdb = .ownclu [clu$l_fstfdb] ; ! GET FIRST FILE DESCRIPTOR BLOCK
          signal (lin$ confbasadr, 5
                  ,cludesc [clu$base_namlng]
                  ,.curhdrisd [isd$v_vpg]^9, lnk$gl_curfil [fdb$g_filename]
                  ,.cludesc [clu$l_base], ownfdb [fdb$g_filename]
                  ,lin$_noimgfil
                  );
          lnk$gl_ctlmsk [lnk$v_image] = false ; ! DON'T MAKE A NON-RUNNABLE IMAGE
        end
      else begin
        if not .lnk$gl_curclu [clu$v_usrbased]
        then begin
          if not .cludesc [clu$v_based] ! IF CLUSTER NOT ALREADY BASED
          then
            lnk$gw_shriscts = .lnk$gw_shriscts - .cludesc [clu$l_nisects] ;
            cludesc [clu$v_based] = true ; ! FLAG AS BASED
            cludesc [clu$l_base] = .curhdrisd [isd$v_vpg]^9 ;
          end ;
        end ;
      end ;
    end ;
```

```
441      0937 7           end ;  
442      0938 7           end  
443      0939 6           else begin  
444      0940 7           if .lnk$gl_curclu [clu$v_based] or .lnk$gl_curclu [clu$v_usrbased]  
445      0941 7           then .lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +  
446      0942 7           .curhdrisd [isd$w_pagcnt] * 512 ;  
447      0943 7           end ;  
448      0944 7           end ;  
449      0945 7           end ;  
450      0946 6           end ;  
451      0947 5           end ;  
452      0948 5           if not .curhdrisd [isd$v_gbl]  
453      0949 5           then begin  
454      0950 5           lnk$alloblk (isd$c_size, curisd) ;  
455      0951 5           curisd [isl$l_nxtisid] = 0 ;  
456      0952 6           curisd [isl$l_bufadr] = 0 ;  
457      0953 6           curisd [isl$l_bufend] = 0 ;  
458      0954 6           curisd [isl$l_cludsc] = .lnk$gl_curclu ;  
459      0955 6           newhdrisd = curisd [isl$t_hdrisd] ;  
460      0956 6           ch$copy (.curhdrisd [isd$w_size]  
461      0957 6           , curhdrisd [isd$w_size], 0  
462      0958 6           , isd$k_maxenglbl, curisd [isl$t_hdrisd]  
463      0959 6           );  
464      0960 6           if .lnk$gl_curclu [clu$l_fstisid] eql 0  
465      0961 6           then ! IF THIS IS THE FIRST  
466      0962 6           begin  
467      0963 6           if .lnk$gl_curclu [clu$v_usrbased]  
468      0964 6           then ! IF BASED BY USER  
469      0965 7           begin  
470      0966 7           if .lnk$gl_curclu [clu$v_based] ! IF CLUSTER IS ALSO BASED  
471      0967 7           then begin  
472      0968 8           if .lnk$gl_curclu [clu$v_based]  
473      0969 8           and .lnk$gl_curclu [clu$l_base] neq .lnk$gl_curclu [clu$l_usrbase]  
474      0970 8           then begin  
475      0971 8           signal ( lin$confbasadr, 5  
476      0972 8           , .lnk$gl_curclu [clu$b_nam$ng]  
477      0973 9           , .lnk$gl_curclu [clu$l_base], .lnk$gl_curfil [fdb$q_filename]  
478      0974 9           , .lnk$gl_curclu [clu$l_usrbase], $descriptor ('Options file')  
479      0975 9           , lin$noimgfil  
480      0976 9           );  
481      0977 9           lnk$gl_ctlmsk [lnk$v_image] = false ;  
482      0978 9           end ;  
483      0979 9           lnk$gl_ctlmsk [lnk$v_image] = false ;  
484      0980 9           lnk$gl_ctlmsk [lnk$v_image] = false ;  
485      0981 8           end ;  
486      0982 8           lnk$gl_ctlmsk [lnk$v_image] = false ;  
487      0983 8           lnk$gl_ctlmsk [lnk$v_image] = false ;  
488      0984 8           lnk$gl_ctlmsk [lnk$v_image] = false ;  
489      0985 8           lnk$gl_ctlmsk [lnk$v_image] = false ;  
490      0986 8           end ;  
491      0987 7           else begin  
492      0988 8           if .lnk$gl_curclu [clu$v_based] ! IF BASED DUE TO ANOTHER IMAGE  
493      0989 8           and not .lnk$gl_curclu [clu$v_pic]  
494      0990 8           then ! BASING IT AND IT TURNED OUT TO  
495      0991 8           begin ! BE NON-PIC  
496      0992 8           begin  
497      0993 9           begin
```

```
: 498      0994  9
: 499      0995  9
: 500      0996 10
: 501      0997 10
: 502      0998 10
: 503      0999 10
: 504      1000 10
: 505      1001 10
: 506      1002 10
: 507      1003 10
: 508      1004 10
: 509      1005 10
: 510      1006 10
: 511      1007  9
: 512      1008  8
: 513      1009  8
: 514      1010  8
: 515      1011  9
: 516      1012  9
: 517      1013  9
: 518      1014  8
: 519      1015  7
: 520      1016  7
: 521      1017  6
: 522      1018  6
: 523      1019  6
: 524      1020  6
: 525      1021  6
: 526      1022  7
: 527      1023  7
: 528      1024  7
: 529      1025  7
: 530      1026  7
: 531      1027  7
: 532      1028  6
: 533      1029  6
: 534      1030  6
: 535      1031  6
: 536      1032  6
: 537      1033  7
: 538      1034  7
: 539      1035  7
: 540      1036  7
: 541      1037  7
: 542      1038  7
: 543      1039  7
: 544      1040  7
: 545      1041  7
: 546      1042  7
: 547      1043  7
: 548      1044  6
: 549      1045  6
: 550      1046  6
: 551      1047  6
: 552      1048  6
: 553      1049  6
: 554      1050  6

        if .Lnk$gl_curclu [clu$l_base] neq .newhdrisd [isd$v_vpg]^9
        then
            begin
                ownclu = .Lnk$gl_curclu [clu$l_ownclu];
                ownfdb = .ownclu [clu$l_fstfdb];
                signal ( lin$ confbasadr, 5
                        , .Lnk$gl_curclu [clu$b_namlng]
                        , .newhdrisd [isd$v_vpg]^9, .Lnk$gl_curfil [fdb$g_filename]
                        , .Lnk$gl_curclu [clu$l_base], ownfdb [fdb$g_filename]
                        );
                lin$_noimgfil
            ;
            Lnk$gl_ctlmsk [Lnk$v_image] = false;
        end
    else
        begin
            if not .Lnk$gl_curclu [clu$v_pic] ! THEN EXTRACT BASE VPN
            then
                begin
                    Lnk$gl_curclu [clu$l_base] = .newhdrisd [isd$v_vpg]^9;
                    Lnk$gl_curclu [clu$v_based] = true;
                end;
            end;
        firstisd = .newhdrisd; ! POINT TO FIRST ISECT IN CLUSTER
    end;
    Lnk$gl_curclu [clu$l_nisects] = .Lnk$gl_curclu [clu$l_nisects] + 1; ! COUNT ISECT IN CLUSTER
    begin
        bind lastisd = Lnk$gl_curclu [clu$l_lstisd] ! POINTER TO LAST ISD IN CLUSTER
        : ref block [, byte];
        lastisd [isl$l_nxtisd] = .curisd; ! PUT AT END OF LIST
        lastisd = .curisd; ! AND MAKE CURRENT THE NEW LAST
    end;

    if .Lnk$gl_curclu [clu$v_pic] ! IF A PIC CLUSTER
    then
        begin
            if .newhdrisd eql .firstisd ! IF FIRST ISECT THIS CLUSTER
            then
                firstisdvpg = .newhdrisd [isd$v_vpg]; ! THEN SAVE VPG OF FIRST ISECT
            newhdrisd [isd$v_vpg] = .newhdrisd [isd$v_vpg] - .firstisdvpg; ! THEN SUBTRACT OUT THE BASE
            : for later re-location (needed) ! FOR OLD IMAGES LINKED @200
            if not .Lnk$gl_curclu [clu$v_based] ! COUNT IT
            then Lnk$gw_shriscts = .Lnk$gw_shriscts + 1;
            Lnk$gl_curclu [clu$l_pages] = .Lnk$gl_curclu [clu$l_pages] + .newhdrisd [isd$w_pagcnt]; ! ACCUMULATE THE PAGES REQUIRED
        end;
    if (ownclu = .Lnk$gl_curclu [clu$l_ownclu]) eql 0
    then
```

```

555      1051 6
556      1052 6
557      1053 6
558      1054 6
559      1055 6
560      1056 6
561      1057 6
562      1058 6
563      1059 6
564      1060 6
565      1061 6
566      1062 6
567      1063 6
568      1064 6
569      1065 6
570      1066 6
571      1067 6
572      1068 6
573      1069 6
574      1070 6
575      1071 6
576      1072 6
577      1073 6
578      1074 6
579      1075 6
580      1076 7
581      1077 7
582      1078 7
583      1079 7
584      1080 7
585      1081 7
586      1082 7
587      1083 7
588      1084 7
589      1085 7
590      1086 7
591      1087 7
592      1088 7
593      1089 7
594      1090 8
595      1091 8
596      1092 8
597      1093 8
598      1094 8
599      1095 8
600      1096 7
601      1097 7
602      1098 7
603      1099 7
604      1100 7
605      1101 7
606      1102 7
607      1103 7
608      1104 7
609      1105 7
610      1106 7
611      1107 7

      lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
      .newhdrisd [isd$w_pagcnt]*512
else
  ownclu [clu$l_cluoff] = .ownclu [clu$l_cluoff] + .newhdrisd [isd$w_pagcnt]*512 ;
  if .curhdrisd [isd$v_wrt]
    and not
    .curhdrisd [isd$v_crf]
  then
    lnk$gl_curclu [clu$v_wrt] = true ;
    ! THEN REMEMBER FOR LNKIMGOUT
  if .lnk$gl_curclu [clu$v_based]
  then
    newhdrisd [isd$v_based] = true ;
    ! IN CLUSTER DESCRIPTOR ALSO
  if not .curhdrisd [isd$v_dzro]
    and .curhdrisd [isd$l_vbn] neq 0
  then
    if .curhdrisd [isd$l_vbn] gequ .maxisdvbn
    then
      maxisdvbn = .curhdrisd [isd$l_vbn] +
      .curhdrisd [isd$w_pagcnt] ;
    ! IF IMAGE SECTION GOES BEYOND LAST
    ! BLOCK OF LAST ISECT, THEN
    ! COMPUTE NEW MAX VBN IN USE
  if not .curhdrisd [isd$v_gbl]
  then
    begin
      local gblsect_namlng ;
      gblsect_namlng = .lnk$gl_curclu [clu$b_namlng] + suffix_size ;
      ! The size of this global isd = length of private isd
      ! + length of gblsect ident
      ! + length of gblsect name count byte
      ! + length of gblsect name
      newhdrisd [isd$w_size] = isd$k_lenpriv + .gblsect_namlng + 5 ; ! SET SIZE AND
      newhdrisd [isd$v_gbl] = true ; ! MAKE IT GLOBAL NOW
      ! GLOBAL ISDS
      ! COMPUTE ISD NAME BY
      do begin
        if (curisdseq [.i] = .curisdseq [.i] + 1) gtru %c'9' ! INCREMENTING THE SUFFIX
        then
          curisdseq [.i] = %c'0'
        else
          exitloop ;
      end ;
      (newhdrisd [isd$t_gblnam])<0, 8, 0> = .gblsect_namlng ;
      ! COPY THE CLUSTER
      ! NAME CONCATENATED WITH
      ch$copy (.lnk$gl_curclu [clu$b_namlng]
      ,lnk$gl_curclu [clu$t_name], 4
      ,curisdseq [0], 0, .gblsect_namlng
      ,newhdrisd [isd$t_gb[nam]]+1
      ! FILL THEN SET MATCH CONTROL
      newhdrisd [isd$v_matchctl] = .lnk$gl_curclu [clu$v_matchctl] ;

```

```

612      1108 7      newhdrisd [isd$l_ident] = .isectident ; ! AND THE MATCH CONTROL IDENT
613      1109 6      end ;
614      1110 6
615      1111 6      if not .curhdrisd [isd$v_dzro]
616      1112 6      and .lnk$gl_curclu [clu$v_copy]
617      1113 6      and .curhdrisd [isd$l_vbn] neq 0
618      1114 6      then begin
619      1115 7          if .curhdrisd [isd$w_pagcnt] gtr .lnk$gl_gsbudsc [0] ! MAXIMIZE THE SIZE OF
620      1116 7          then .lnk$gl_gsbudsc [0] = .curhdrisd [isd$w_pagcnt] ; ! BUFFER WE'LL NEED
621      1117 7
622      1118 7
623      1119 6      end ;
624      1120 6
625      1121 6      if .iafva neq 0 ! IF THERE IS A FIXUP SECTION
626      1122 6      and ! AND THIS IS IT
627      1123 7      (.newhdrisd [isd$v_vpg]^9 eql .iafva)
628      1124 6      then begin
629      1125 7          .lnk$gl_curclu [clu$l_fixisd] = .curisd ; ! THEN REMEMBER IT FOR LATER
630      1126 7          newhdrisd [isd$v_fixupvec] = true ; ! FLAG FIXUP SECTION IN ISD
631      1127 7          end ;
632      1128 6
633      1129 6
634      1130 6      .lnk$gw_nisects = .lnk$gw_nisects + 1 ; ! COUNT THAT IMAGE SECTION
635      1131 5      end ; ! OF LOCAL ISECT
636      1132 5
637      1133 6      if (curhdrisd = .curhdrisd + .nxtisoff) gtr (.header + 510) ! CHECK IT WAS COMPLETELY
638      1134 5      then signal_stop (lin$_badimghdr, 1 ! CONTAINED BY THE CURRENT BLOCK
639      1135 5          ; .lnk$gl_curfil [fdb$q_filename] ! IF NOT TRUE, FATAL IMAGE
640      1136 5          ) ; ! HEADER ERROR
641      1137 5      end ! END OF ISECTION LOOP
642      1138 5
643      1139 4      else ! IMAGE SECTION WAS A STACK ISD, JUST SKIP IT BUT MAKE SURE IT IS CONTAINED BY THE
644      1140 4          CURRENT BLOCK -- ISSUE ERROR AND QUIT IF NOT
645      1141 4
646      1142 4
647      1143 4
648      1144 5      if (curhdrisd = .curhdrisd + .nxtisoff) gtr (.header + 510)
649      1145 4      then signal_stop (lin$_badimghdr, 1, .lnk$gl_curfil [fdb$q_filename]) ;
650      1146 4
651      1147 4
652      1148 3      end ; ! END OF BLOCK LOOP
653      1149 2      end ;
654      1150 2
655      1151 2      if .maxisdvbn gtr .gstvbn [0] or .hdrblkcnt neq 0 ! AND THIS SHOULD ALSO POINT TO GST
656      1152 2          ! AND NO HEADER BLOCKS REMAIN
657      1153 2      then signal_stop (lin$_badimghdr, 1, .lnk$gl_curfil [fdb$q_filename]) ; ! IF EITHER ABOVE NOT TRUE,
658      1154 2          ! FATAL IMAGE HEADER ERROR
659      1155 2
660      1156 2      if .lnk$gl_curclu [clu$v_based] ! IF THIS CLUSTER IS BASED
661      1157 2          and .lnk$gl_defclu [clu$v_based] ! AND BASE= IN OPTION FILE
662      1158 2          and .lnk$gl_curclu [clu$l_base] lequ .lnk$gl_defclu [clu$l_base] ! AND THIS IMAGE IS BELOW IT
663      1159 2      then signal (lin$_basshrbel, 3, .lnk$gl_curfil [fdb$q_filename] ! THEN WARN USER SHR IMG
664      1160 2          ; .lnk$gl_curclu [clu$l_base], .lnk$gl_defclu [clu$l_base] ! BELOW BASE=
665      1161 2          ) ;
666      1162 2
667      1163 2
668      1164 2      gstvbn [0] = .gstvbn [0] + .blockoffset ; ! RELOCATE DOWN FILE IF NECESSARY

```

```

669 1165 2 gstvbn [1] = 0 ;           ! CLEAR THE BYTE OFFSET IN THE RFA
670 1166 2 saverecount = .Lnk$gl_objrecs ;   ! SAVE CURRENT RECORD COUNT
671 1167 2 Lnk$pointobj (gstvbn) ;       ! POINT TO GST
672 1168 2
673 1169 2 if not Lnk$procsobj (gstvbn)    ! AND GO PROCESS IT LIKE AN
674 1170 2 then                           ! OBJECT RETURNING IF ERROR
675 1171 2 return false ;                 ! GET NUMBER PROCESSED
676 1172 2
677 1173 2 saverecount = .Lnk$gl_objrecs - .saverecount ;   ! AND IF NOT CORRECT NUMBER
678 1174 2
679 1175 2 if .saverecount neq .gstrecs    ! FATAL IMAGE HEADER ERROR
680 1176 2 then
681 1177 2     signal_stop (lin$_badimghdr, 1, Lnk$gl_curfil [fdb$g_filename]) ;
682 1178 2
683 1179 2 Lnk$gl_imgrecs = .Lnk$gl_imgrecs + .saverecount ;   ! ACCUMULATE RECORD COUNT
684 1180 2 return true ;                  ! ALL DONE SO RETURN SUCCESS
685 1181 1 end ;                      ! End of LNK$PROCSHRIM

```

```

.TITLE LNK_PROCSHRIM
.IDENT \V04-000\
.PSECT SPLIT$,NOWRT,NOEXE,2
65 6C 69 66 20 73 6E 6F 69 74 70 4F 00000 P.AAB: .ASCII \Options file\
0000000C 0000C P.AAA: .LONG 12
00000000 00010 .ADDRESS P.AAB
:P
.PSECT SOWNS,NOEXE,2
30 30 30 5F 00000 CURISDSEQ: .ASCII \_000\
00004 HDRBLKCNT: .BLKB 4
00008 HEADERBLOCK: .BLKB 4
:P
.PSECT $GLOBAL$,NOEXE,2
00000 LNK$GL_GSBUFDESC:: .BLKB 12
0000C LNK$GL_IMGRECS:: .BLKB 4
:I
ISDSC_SIZE== 88
HDRSK_FILLCHR== 255
IHDSK_SHR== 2
IHDSK_ACTIVOFF== 48
IHDSK_SYMDBGOFF== 68
IHDSK_IMGIDOFF== 88
IHDSK_PATCHOFF== 168
IHDSK_MAXLENGTH== 168
HDRSK_MINFILL== 2
:EXTRN LNK$ALLOBLK, LNK$ALLOCCLUSTER
:EXTRN LNK$CLOSEFILE, LNK$POINTOBJ
:EXTRN LNK$ADDIMAGE, LNK$PROCSOBJ
:EXTRN LIN$_BADIMGHDR, LIN$_BASSHRBEL
:
```

LN  
VO

```

        .EXTRN LINS_CONFBA$ADR
        .EXTRN LINS_IDMISMCH, LINS_IMGBASED
        .EXTRN LINS_NOIMGFIL, LINS_NONPICIMG
        .EXTRN LINS_RELINK, LINS_READERR
        .EXTRN LNK$GL_SHRCLSTRS
        .EXTRN LNK$AL_RAB, LNK$GW_NISECTS
        .EXTRN LNK$GW_SHRISCTS
        .EXTRN LNK$GB_PASS, LNK$GL_CURFIL
        .EXTRN LNK$GL_CURCLU, LNK$GL_DEFCLU
        .EXTRN LNK$GL_LASTCLU, LNK$GE_CTLMSK
        .EXTRN LNK$GL_OBJRECS

        .PSECT SCODE$, NOWRT, 2

        .ENTRY LNK$PROCSHRIM, Save R2,R3,R4,R5,R6,R7,R8,- ; 0652
               R9 R10 R11
               -88(SP), SP
               TSTB (AP)
               BNEQ 1$
               CLRL BLOCKOFFSET
               BRB 2$
               SUBL3 #1, @MODRFA, BLOCKOFFSET
               INCL LNK$GL_SHRCLSTRS
               CLRL CLUDESC
               MOVL #1, HDRBLKCNT
               MNEGL #1, NXTISDOFF
               CLRL MAXISDVBN
               MOVL BLOCKOFFSET, HEADERBLOCK
               INSV #3158064, #0, #24, CURISDSEQ+1
               MOVAB 1(R9), 44(SP)
               CMPL NXTISDOFF, #-1
               BEQL 4$
               BRW 59$
               CALLS #0, READNEXTBLOCK
               BLBS R0, 5$
               ADDL3 #20, LNK$GL_CURFIL, -(SP)
               PUSHL #1
               PUSHL #LINS_BADIMGHDR
               CALLS #3, LIBSTOP
               MOVL HEADER, R0
               CMPL HEADERBLOCK, 44(SP)
               BEQL 6$
               BRW 17$
               CMPB 17(R0), #2
               BNEQ 9$
               CMPW 12(R0), #12848
               BNEQ 9$
               CMPB 14(R0), #48
               BNEQ 7$
               CMPB 15(R0), #53
               BGTRU 9$
               MOVZWL 8(R0), R1
               ADDL2 #44, R1
               CMPL R1, #168
               BGEQU 8$
               MOVZBL #168, R1
               CMPZV #0, #16, (R0), R1

```

:





59	A2	00000000G	00	54	AE	9F	002B5	PUSHAB	SHRDESC				
		50	00000000G	00	04	FB	002B8	CALLS	#4, LNK\$ADDIMAGE				0872
		51	58	A0	00	DO	002BF	MOVL	LNK\$GL_CURCLU, R0				0869
		52	40	AE	00	D0	002CA	MOVAB	88(R0), R1				
		34	3C	AE	00	E8	002D0	MOVL	CLUDESC, R2				
		00	01	A1	00	F0	002D4	BEQL	28\$				
		54	A2	50	00	D0	002DB	BLBS	FOUND, 28\$				0872
		03	61	03	00	E1	002DF	INSV	1(R1), #0, #1, 89(R2)				0874
		58	A2	31	00	E3	002E3	MOVL	R0, 84(R2)				0876
		0C	14	01	00	88	002E6	BBC	#3, (R1), 25\$				0879
		A6	BE	09	00	E1	002EA	BISB2	#1, 88(R2)				0880
		53	17	00	00	EF	002EF	BBC	#9, @20(SP), 26\$				0881
		53	53	09	00	78	002F5	EXTZV	#0, #23, 4(CURHDRISD), R3				
		53	4C	06	00	11	002F9	ASHL	#9, R3, R3				0883
		4C	A0	20	00	C1	002FB	BRB	27\$				0884
		53	A2	53	00	D0	00301	ADDL3	32(R0), 76(R0), R3				0885
		04	04	00B7	00	31	00305	MOVL	R3, 76(R2)				0886
		D0	61	61	00	E9	00308	BRW	35\$				0891
		61	61	03	00	E1	0030B	BLBC	(R1), 29\$				0893
		6E	54	0A	00	E1	0030F	BBC	#3, (R1), 30\$				0895
				54	00	D0	00313	BBC	#10, (R1), 24\$				0898
				54	00	A2	00313	MOVL	84(R2), OWNCLU				0900
				54	00	D4	00317	CLRL	R4				
				6E	00	D5	00319	TSTL	OWNCLU				
				0E	00	13	0031B	BEQL	31\$				
				54	00	D6	0031D	INCL	R4				
		53	53	02	00	A6	0031F	MOVZWL	2(CURHDRISD), R3				0903
			20	53	00	09	78	00323	ASHL	#9, R3, R3			
				A0	00	53	C0	00327	ADDL2	R3, 32(R0)			
				60	00	54	E9	0032B	BLBC	R4, 32\$			0905
		50	6E	00000058	00	8F	C1	0032E	ADDL3	#88, OWNCLU, R0			0907
		55	55	55	00	60	E9	00336	BLBC	(R0), 32\$			
		53	53	4C	00	EF	00339	EXTZV	#0, #23, 4(CURHDRISD), R3				0909
				53	00	09	78	0033F	ASHL	#9, R3, R3			
				53	00	A2	D1	00343	CMPL	76(R2), R3			
				53	00	45	13	00347	BEQL	32\$			
		50	6E	18	00	C1	00349	ADDL3	#8, OWNCLU, R0				0914
				AE	00	60	D0	0034D	MOVL	(R0), OWNFDB			0918
		50	1C	AE	00000000G	8F	DD	00351	PUSHL	#LINS NOIMGFIL			
					14	C1	00357	ADDL3	#20, OWNFDB, R0				
					50	DD	0035C	PUSHL	R0				
		7E	00000000G	00	4C	A2	DD	0035E	PUSHL	76(R2)			
		A6	17	00	14	C1	00361	ADDL3	#20, LNK\$GL CURFIL, -(SP)				0917
		7E	50	00	00	EF	00369	EXTZV	#0, #23, 4(CURHDRISD), R0				
				5C	00	09	78	0036F	ASHL	#9, R0, -(SP)			
					05	A2	9F	00373	PUSHAB	92(R2)			
					05	DD	00376	PUSHL	#5				0916
		00000000G	00	00000000G	00	8F	DD	00378	PUSHL	#LINS CONFBASADR			0918
		00000000G	00	00000000G	00	08	FB	0037E	CALLS	#8, LIB\$SIGNAL			
					01	8A	00385	BICB2	#1, LNK\$GL_CTLMSK				0921
		39	61	0A	00	3D	11	0038C	BRB	36\$			
		08	58	A2	00	E0	0038E	32\$:	BBS	#10, (R1), 36\$			0925
		00000000G	00	48	00	E8	00392	BLBS	88(R2), 33\$				0928
		58	A2	A2	00	A2	00396	SUBW2	72(R2), LNK\$GW_SHRISCTS				0930
		58	A2	01	00	88	0039E	BISB2	#1, 88(R2)				0932
		58	A2	17	00	EF	003A2	EXTZV	#0, #23, 4(CURHDRISD), R0				0933

4C	A2	50	50	09	78	003AB	ASHL	#9	R0,	76(R2)		0891	
		50	00000000G	1C	11	003AD	BRB	36\$				0941	
OC	59	05	58	00	D0	003AF	MOVL	LNK\$GL_CURCLU,	R0				
		A0	02	A0	E8	003B6	BLBS	88(R0),	35\$				
53	20	53	02	A6	F1	003BA	BBC	#2,	89(R0),	36\$		0944	
		53	09	3C	003BF	35\$:	MOVZWL	2(CURHDRISD),	R3				
		A0	53	C0	003C7		ASHL	#9,	R3,	R3			
		03	14	BE	E9	003CB	ADDL2	R3,	32(R0)			0950	
			0262	31	003CF	36\$:	BLBC	@20(SP),	37\$				
			44	AE	9F	003D2	PUSHAB	CURISD				0953	
		7E	58	8F	9A	003D5	MOVZBL	#88,	-(SP)				
		00	02	FB	003D9		CALLS	#2,	LNK\$ALLOBLK			0954	
		5B	44	AE	D0	003E0	MOVL	CURISD,	R11				
			6B	D4	003E4		CLRL	(R11)				0955	
			08	AB	7C	003E6	CLRQ	8(R11)				0957	
0040	8F	10	5A	00000000G	00	D0	003E9	MOVL	LNK\$GL_CURCLU,	R10			
			AB	5A	D0	003F0	MOVL	R10,	18(R11)			0958	
			57	18	AB	003F4	MOVAB	24(R11),	NEWHDRISD			0959	
		00	66	66	2C	003F8	MOVCS	(CURHDRISD),	(CURHDRISD),	#0,	#64,	24(R11)	0961
			18	AB	003FF		TSTL	24(R10)				0963	
			18	AA	D5	00401	BEQL	38\$					
				03	13	00404	BRW	43\$					
		52	51	58	00C8	31	00406	MOVAB	88(R10),	R1			0966
			61	AA	9E	00409	38\$:	BBC	#10,	(R1),	40\$		
			3A	61	0A	0040D	BLBC	(R1),	39\$			0969	
		3C	AA	4C	E1	00411	CMPL	76(R10),	60(R10)			0971	
				AA	D1	00414	BEQL	39\$					
				33	13	00419	PUSHL	#LINS_NOIMGFIL				0976	
				00000000G	8F	DD	0041B	PUSHAB	P.AAA			0977	
				00000000	EF	9F	00421	PUSHL	60(R10)				
		7E	00000000G	3C	AA	DD	00427	ADDL3	#20,	LNK\$GL_CURFIL,	- (SP)		0976
				14	C1	0042A	PUSHL	76(R10)				0975	
				4C	AA	DD	00432	PUSHL	92(R10)			0976	
				5C	AA	9F	00435	PUSHL	#5			0976	
				05	DD	00438	PUSHL	#LINS_CONFBAADR					
			00000000G	8F	DD	0043A	CALLS	#8,	LIB\$SIGNAL			0980	
			00000000G	08	FB	00440	BICB2	#1,	LNK\$GL_CTLMSK			0983	
			00	01	8A	00447	MOVL	LNK\$GL_CURCLU,	R0				
			50	50	00000000G	00	D0	0044E	39\$:	60(R0),	76(R0)		0984
			4C	A0	3C	A0	D0	00455	CLRL	60(R0)			0985
				3C	A0	D4	0045A	BISB2	#1,	88(R0)			0986
			58	A0	01	88	0045D	BRB	42\$				0989
					6A	11	00461	BLBC	(R1),	41\$			0991
				55	61	E9	00463	40\$:	BBS	#3,	(R1),	42\$	0994
		04	63	61	03	E0	00466	EXTZV	#0,	#23,	4(NEWHDRISD),	R0	
			A7	17	00	EF	0046A	ASHL	#9,	R0,	R0		
			50	50	09	78	00470	CMPL	76(R10),	R0			
				4C	AA	D1	00474	BEQL	42\$				
					53	13	00478	MOVL	84(R10),	OWNCLU			0997
		52	6E	54	AA	D0	0047A	ADDL3	#8,	OWNCLU,	R2		0998
			6E	08	C1	0047E	MOVL	(R2),	OWNFDB				
			18	AE	62	D0	00482	PUSHL	#LINS_NOIMGFIL			1002	
			00000000G	8F	DD	00486	ADDL3	#20,	OWNFDB,	R1			
		51	1C	AE	14	C1	0048C	PUSHL	R1				
					51	DD	00491	PUSHL	76(R10)				
				4C	AA	DD	00493	PUSHL					

7E 00000000G 00 14 C1 00496 ADDL3 #20, LNK\$GL\_CURFIL, -(SP) : 1001  
                       5C 50 DD 0049E PUSHL R0 : 1002  
                       AA 9F 004A0 PUSHAB 92(R10) : 1000  
                       05 DD 004A3 PUSHL #5 : 1002  
                       8F DD 004A5 PUSHL #LINS\_CONFBAADR :  
                       08 FB 004AB CALLS #8, LIBSSIGNAL :  
                       01 8A 004B2 BICB2 #1, LNK\$GL\_CTLMSK : 1005  
                       12 11 004B9 BRB 42\$ : 0993  
                       03 E0 004BB 41\$: BBS #3, (R1), 42\$ : 1009  
                       00 EF 004BF EXTZV #0, #23, 4(NEWHDRISD), R0 : 1012  
                       09 78 004C5 ASHL #9, R0, 76(R10) :  
                       01 88 004CA BISB2 #1, (R1) : 1013  
                       57 D0 004CD 42\$: MOVL NEWHDRISD, FIRSTISD : 1016  
                       00 D0 004D1 43\$: MOVL LNK\$GL\_CURCLU, R10 : 1019  
                       AA D6 004D8 INCL 72(R10) :  
                       5B D0 004DB MOVL R11, @28(R10) : 1026  
                       5B D0 004DF MOVL R11, 28(R10) : 1027  
                       AA 9E 004E3 MOVAB 88(R10), 4(SP) : 1031  
                       03 E1 004E8 BBC #3, @4(SP), 45\$ :  
                       57 D1 004ED CMPL NEWHDRISD, FIRSTISD : 1034  
                       07 12 004F1 BNEQ 44\$ :  
                       00 EF 004F3 EXTZV #0, #23, 4(NEWHDRISD), FIRSTISDVPG : 1036  
                       00 EF 004FA 44\$: EXTZV #0, #23, 4(NEWHDRISD), R0 : 1039  
                       50 F0 00500 SUBL2 FIRSTISDVPG, R0 :  
                       00 B6 0050E INSV R0, #0, #23, 4(NEWHDRISD) :  
                       50 02 A7 3C 00514 45\$: BLBS @4(SP), 45\$ : 1042  
                       AA 50 CO 00518 INCW LNK\$GW\_SHRISCTS : 1043  
                       6E 54 AA D0 0051C MOVZWL 2(NEWHDRISD), R0 : 1046  
                       06 04 BE E8 0050A ADDL2 R0, 80(R10) :  
                       00 00000000G 00 B6 0050E MOVL 84(R10), OWNCLU : 1049  
                       50 02 A7 3C 00522 BNEQ 46\$ :  
                       50 02 09 78 00526 MOVZWL 2(NEWHDRISD), R0 : 1052  
                       50 50 CO 0052A ASHL #9, R0, R0 :  
                       20 AA 0F 11 0052E ADDL2 R0, 32(R10) :  
                       09 78 00526 BRB 47\$ :  
                       50 02 A7 3C 00530 46\$: MOVZWL 2(NEWHDRISD), R0 : 1054  
                       50 09 78 00534 ASHL #9, R0, R0 :  
                       51 6E 20 C1 00538 ADDL3 #32, OWNCLU, R1 :  
                       61 50 CO 0053C ADDL2 R0, (R1) :  
                       0A 14 BE 03 E1 0053F 47\$: BBC #3, @20(SP), 48\$ : 1056  
                       05 14 BE 01 E0 00544 BBS #1, @20(SP), 48\$ : 1058  
                       04 BE 40 8F 88 00549 BISB2 #64, @4(SP) : 1060  
                       04 04 BE 04 E9 0054E 48\$: BLBL @4(SP), 49\$ : 1062  
                       16 09 A7 02 88 00552 BISB2 #2, 9(NEWHDRISD) : 1064  
                       14 BE 02 E0 00556 49\$: BBS #2, @20(SP), 50\$ : 1066  
                       OC A6 D5 0055B TSTL 12(CURHDRISD) : 1067  
                       24 AE OC A6 D1 00560 BEQL 50\$ :  
                       OC A6 0A 1F 00565 CMPL 12(CURHDRISD), MAXISDVBN : 1069  
                       24 50 02 A6 3C 00567 BLSSU 50\$ :  
                       AE 0C B640 9E 0056B MOVZWL 2(CURHDRISD), R0 : 1072  
                       79 14 BE E8 00571 50\$: MOVAB @12(CURHDRISD)[R0], MAXISDVBN :  
                       50 5C AA 9A 00575 BLBS @20(SP), 54\$ : 1074  
                       50 04 CO 00579 MOVZBL 92(R10), GBLSECT\_NAMLNG : 1079  
                       50 15 A1 0057C ADDL2 #4, GBLSECT\_NAMLNG :  
                       67 08 A7 01 88 00580 ADDW3 #21, GBLSECT\_NAMLNG, (NEWHDRISD) : 1086  
                       51 03 D0 00584 BISB2 #1, 8(NEWHDRISD) : 1087  
                       03 D0 00584 MOVL #3, I : 1089



00000000G	00	4C	A0	D1	0069B	CMPL	76(R0), LNK\$GL_DEFCLU+76	1158	
			20	1A	006A3	BGTRU	62\$		
		00000000G	00	DD	006A5	PUSHL	LNK\$GL_DEFCLU+76	1161	
		4C	A0	DD	006AB	PUSHL	76(R0)		
7E	00000000G	00	14	C1	006AE	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	1160	
			03	DD	006B6	PUSHL	#3		
		00000000G	8F	DD	006B8	PUSHL	#LINS_BASSHRBEL		
		50	00	FB	006BE	CALLS	#5, LIBSSIGNAL		
		AE	59	C0	006C5	62\$: ADDL2	BLOCKOFFSET, GSTVBN	1164	
			54	AE	D4	006C9	CLRL	GSTVBN+4	1165
		52	00000000G	00	DD	006CC	MOVL	LNK\$GL_OBJRECS, SAVERECOUNT	1166
			50	AE	9F	006D3	PUSHAB	GSTVBN	1167
		00000000G	00	01	FB	006D6	CALLS	#1, LNK\$POINTOBJ	
			50	AE	9F	006DD	PUSHAB	GSIVBN	
		00000000G	00	01	FB	006E0	CALLS	#1, LNK\$PROCSOBJ	1169
		30	50	E9	006E7	BLBC	R0, 64\$		
52	00000000G	00	52	C3	006EA	SUBL3	SAVERECOUNT, LNK\$GL_OBJRECS, SAVERECOUNT	1173	
		28	AE	52	D1	006F2	CMPL	SAVERECOUNT, GSTRECS	1175
				17	13	006F6	BEQL	63\$	
7E	00000000G	00	14	C1	006F8	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	1177	
			01	DD	00700	PUSHL	#1		
		00000000G	8F	DD	00702	PUSHL	#LINS_BADIMGHDR		
		00000000	00	03	FB	00708	CALLS	#3, LIBSTOP	
		EF	52	C0	0070F	63\$: ADDL2	SAVERECOUNT, LNK\$GL_IMGRECS	1179	
		50	01	DO	00716	MOVL	#1, R0	1180	
				04	00719	RET			
			50	D4	0071A	64\$: CLRL	RO		
				04	0071C	RET		1181	

: Routine Size: 1821 bytes, Routine Base: \$CODE\$ + 0000

: 686 1182 1

```

688   1183 1 routine readnextblock =           ! READ NEXT BLOCK IF ANY
689   1184 1 ++
690   1185 1 THIS ROUTINE IS CALLED TO READ THE NEXT BLOCK OF THE IMAGE HEADER
691   1186 1 READ ERRORS ARE FATAL. A REQUEST TO READ ANOTHER BLOCK
692   1187 1 WHEN HDRBLKCNT IS ALREADY ZERO RETURNS FATAL.
693   1188 1 HDRBLKCNT IS DECREMENTED AFTER EACH READ AND HEADERBLOCK IS
694   1189 1 INCREMENTED BEFORE EACH READ.
695   1190 1
696   1191 1 --
697   1192 2 begin
698   1193 2 local
699   1194 2     saveusz,
700   1195 2     readerror ;
701   1196 2
702   1197 2 if .hdrblkcnt leq 0           ! IF NO MORE BLOCKS
703   1198 2 then                   ! RETURN FAILURE
704   1199 2     return false ;
705   1200 2
706   1201 2 saveusz    = .Lnk$al_rab [rab$w_usz] ;           ! SAVE USZ
707   1202 2 headerblock = .headerblock + 1 ;           ! SET THE BLOCK TO READ
708   1203 2
709   1204 2 Lnk$al_rab [rab$l_bkt] = .headerblock ;           ! SET STARTING VBN
710   1205 2 Lnk$al_rab [rab$w_usz] = 512 ;           ! AND SET THE BYTE COUNT
711   1206 2
712   1207 3 if not (readerror = $read (rab = Lnk$al_rab))           ! ATTEMPT TO READ LIBRARY, USING
713   1208 2 then                   ! BUFFER ALREADY SET UP
714   1209 3     begin
715   1210 3       signal (lin$_readerr, 1
716   1211 3             , Lnk$gl_curfil [fdb$g_filename]
717   1212 3             , .readerror, .Lnk$al_rab [rab$l_stv]
718   1213 3         );
719   1214 3       Lnk$closefile (.Lnk$gl_curfil) ;           ! THE MESSAGES AND ATTEMPT TO
720   1215 3       Lnk$al_rab [rab$w_usz] = .saveusz ;           ! RESTORE USZ
721   1216 3       return false ;
722   1217 2     end ;
723   1218 2
724   1219 2     hdrblkcnt = .hdrblkcnt - 1 ;           ! DECREMENT THE BLOCK COUNT
725   1220 2     Lnk$al_rab [rab$w_usz] = .saveusz ;           ! AND ALL DONE SUCCESSFULLY
726   1221 2     return true ;
727   1222 1 end ;

```

## .EXTRN SYSSREAD

## 003C 00000 READNEXTBLOCK:

				.WORD	Save R2,R3,R4,R5	1183
	55 00000000G	00 9E 00002		MOVAB	LNK\$GL CURFIL, R5	
	54 00000000'	EF 9E 00009		MOVAB	HDRBLKCNT, R4	
	53 00000000G	00 9E 00010		MOVAB	LNKSAL_RAB+32, R3	
		64 D5 00017		TSTL	HDRBLKCNT	1197
		4C 15 00019		BLEQ	2\$	
	52	63 3C 0001B		MOVZWL	LNKSAL_RAB+32, SAVEUSZ	1201
18	A3	04 A4 D6 0001E		INCL	HEADERBLOCK	1202
	63	0200 8F B0 00021		MOVL	HEADERBLOCK, LNK\$AL_RAB+56	1204
		E0 A3 9F 0002B		MOVW	#512, LNK\$AL_RAB+32-	1205
				PUSHAB	LNKSAL_RAB	1207

00000000G	00		01	FB 0002E	CALLS #1, SYSSREAD	
	26		50	E8 00035	BLBS READERROR, 1\$	1212
		EC	A3	DD 00038	PUSHL LNK\$AL_RAB+12	
7E	65		50	DD 0003B	PUSHL READERROR	1211
			14	C1 0003D	ADDL3 #20, LNK\$GL_CURFIL, -(SP)	
			01	DD 00041	PUSHL #1	
	00000000G		8F	DD 00043	PUSHL #LINS READERR	
00000000G	00		05	FB 00049	CALLS #5, LIB\$SIGNAL	1214
			65	DD 00050	PUSHL LNK\$GL_CURFIL	
00000000G	00		01	FB 00052	CALLS #1, LNR\$CLOSEFILE	1215
	63		52	B0 00059	MOVW SAVEUSZ, LNK\$AL_RAB+32	1216
			09	11 0005C	BRB 2\$	1219
			64	D7 0005E	1\$: DECL HDRBLKCNT	
63			52	B0 00060	MOVW SAVEUSZ, LNK\$AL_RAB+32	1220
	50		01	D0 00063	MOVL #1, R0	1221
			04	00066	RET	
			50	D4 00067	2\$: CLRL R0	1222
			04	00069	RET	

: Routine Size: 106 bytes, Routine Base: \$CODE\$ + 071D

: 728 1223 1 end  
: 729 1224 1  
: 730 1225 0 eludom

! End of module

.EXTRN LIB\$SIGNAL, LIB\$STOP

## PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	16	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$OWNS	12	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$SPLITS	20	NOVEC,NOWRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	1927	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
. ABS .	0	NOVEC,NOWRT,NORD, NOEXE,NOSHR, LCL, ABS, CON,NOPIC,ALIGN(0)

## Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	68	0	1000	00:02.0
\$255\$DUA28:[LINKER.OBJ]DATBAS.L32;1	538	41	7	28	00:00.8

LNK\_PROCSHRIM  
V04-000

K 9

16-Sep-1984 00:30:18  
14-Sep-1984 12:40:34

VAX-11 Bliss-32 V4.0-742  
[LINKER.SRC]LNKPROSHR.B32;2

Page 24  
(3)

: COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS\$:LNKPROSHR/OBJ=OBJ\$:LNKPROSHR MSRC\$:\$LNKPROSHR/UPDATE=(ENH\$:\$LNKPROSHR)

: Size: 1927 code + 48 data bytes

: Run Time: 00:50.4

: Elapsed Time: 02:02.6

: Lines/CPU Min: 1457

: Lexemes/CPU-Min: 28711

: Memory Used: 563 pages

: Compilation Complete

0219 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

LNKPRLTB  
LIS

LNKSYMOUT  
LIS

LNKSYMTBL  
LIS

LNKUMALLO  
LIS

LNKPSCTB  
LIS

LNKPROSHR  
LIS

LNKSTATSO  
LIS